A quantitative Comparison of Checkpoint with Restart and Replication in Volatile Environments

Rong Zheng and Jaspal Subhlok
Department of Computer Science
University of Houston
Houston, TX 77204
E-mail: *rzheng@cs.uh.edu*

Department of Computer Science
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

Technical Report Number UH-CS-08-06

June 23, 2008

Keywords: Fault tolerance, checkpoint with restart, replication

Abstract

Volatile computing environments such as desktop grids differs from traditional systems in the high volatility of compute nodes in both reachability and availability of compute resource. As a result, different fault tolerant techniques are required to ensure efficient execution of parallel jobs. This technical report summarizes failure and availability patterns of distributed computing systems; and propose simple models for characterizing the impact of parameters on the efficiency of checkpoint with restart and replication schemes. Our analysis shows that when the number of processors and/or the failure rate are high, it is indeed beneficial to use replication as it renders large speedup in comparison to checkpoint with restart with the optimal parameter settings.

A quantitative Comparison of Checkpoint with Restart and Replication in Volatile Environments

Rong Zheng and Jaspal Subhlok
Department of Computer Science
University of Houston
Houston, TX 77204
E-mail: *rzheng@cs.uh.edu*

Abstract

Volatile computing environments such as desktop grids differs from traditional systems in the high volatility of compute nodes in both reachability and availability of compute resource. As a result, different fault tolerant techniques are required to ensure efficient execution of parallel jobs. This technical report summarizes failure and availability patterns of distributed computing systems; and propose simple models for characterizing the impact of parameters on the efficiency of checkpoint with restart and replication schemes. Our analysis shows that when the number of processors and/or the failure rate are high, it is indeed beneficial to use replication as it renders large speedup in comparison to checkpoint with restart with the optimal parameter settings.

Index Terms

Fault tolerance, checkpoint with restart, replication

I. Failure and availability pattern of distributed computing systems

Design of dependable computing facilities requires a thorough understanding of the failure patterns of real systems. Unfortunately, the failure behaviors e.g., mean time before failure (MTBF), mean time to recovery (MTTR), correlations among failures of different components, vary drastically from system to system and are constantly evolving with the evolution of new hardware and software architecture. The issue of obtaining a proper set of realistic parameters is further complicated by soft failures, where a host machine is physically operational but is not able to carry out computational tasks.

In this section, we summarize existing measurement based studies that characterize the failure and availability patterns. Broadly, distributed computing systems are divided into three categories: high performance computing systems, campus/enterprise desktop grids, wide-area volunteer computing systems or P2P systems.

HPC systems: In [11], Schroeder and Gibson analyzed data collected over the past 9 years at Los Alamos National Laboratory and includes 23000 failures recorded on more than 20 different systems, mostly large clusters of SMP and NUMA nodes. They found that average failure rates differ wildly across systems, ranging from 20 – 1000 failures per year (0.25 to 3 failures per year per process), and that time between failures is modeled well by a Weibull distribution with decreasing hazard rate. From one system to another, mean repair time varies from less than an hour to more than a day, and repair times are well modeled by a log-normal distribution. The failure repair time is correlated with the root cause of failures.

Iosup *et al.* [5] analyze the resource availability of Grid'5000, an experimental grid environment of over 2,500 processors. They found that on average, resource availability in Grid'5000 is 69% ($\pm$11.4), with a maximum of 92% and a minimum of 35%. The mean time between failures (MTBF) of the environment is of 744$\pm$2631 seconds, that is around 12 minutes. At a cluster level, resource availability varies from 39% up to 98% across the 15 clusters. The average MTBF for all clusters is 18404$\pm$13848 seconds, so around 5 hours. Similar observations have been made in the study of production grid such as TeraGrid and Geon by Khalili et al. However, the authors point out most of the unavailability is due to middle-ware problems.

Campus/enterprise desktop grids: Kondo *et al.* [6], [7] conduct studies on the resource availability of desktop grids from desktop PCs at the San Diego Super Computer Center and at the University of Paris South. Three types of availabilities are considered, i.e., host availability, CPU availability, and task execution availability. The task execution availability, defined as the percentage of CPU that a task can execute on the host or not, according

to a desktop grid worker's recruitment policy. The authors observed task execution availability ranging from tens of minutes to 23.5 hrs with unavailability interval from 20 mins to 32 hours. Host availability is shown to be on average close to 6 hours during weekends, versus under 3 hours during weekdays. In [9], Nurmi *et al.* evaluate the suitability of four potential statistical distributions exponential, Pareto, Weibull, and hyper-exponential using traces from 8-week data from UCSB computer science student lab of 83 machines, Condor pool running at the University of Wisconsin consisting of 210 machines during a 6-week period, a remote Internet host availability from 1170 machines over a 3 month experimental period. Their study showed that either a hyper-exponential or Weibull model effectively represents machine availability in enterprise and Internet computing environment.

Wide-area computing systems: Availability of wide-area computing systems has also been studied in the P2P networking community. Using crawlers and probers to probe and collect active hosts on the Internet over a period of 7 days, Bhagwan *et al* [1] determined the distribution and correlation of host reachability. A set of availability predictors are devised using traces from Planetlab, Overnet and Microsoft incorporate in [8] to improve replica placement, routing in delay-tolerant network, and forecasts of global infection dynamics. Both work limits the availability to host reachability. The availability of computing and storage resource is not tracked.

II. Modeling checkpoint restart

Modeling the impact of parameters on the efficiency of checkpoint restart has been an active research topic since the 70's. In [13], Young gives a first-order model to analyze the expected total lost time due to failure and checkpointing. His model does not consider the failure repair time on application performance and is based on the assumption that $MTBF$ is significantly larger than checkpoint time. Daly [2]–[4] improved Young's model by relaxing the constraint and proposed a simple high-order approximation that allows evaluation of the optimal checkpoint interval in a closed form. The work by Pattabiraman *et. al* [10] uses Stochastic Activity Networks to model coordinated checkpointing for large-scale supercomputers; and considers synchronization overhead, failures during checkpointing and recovery, and correlated failures. In [12], Wu *et. al* propose to use an M/G/1 process to describe system failures, where the failure process is modeled as Poisson and the recovery time follows a general distribution. The distribution of the application completion time with system size N and work load W is also given.

III. Comparison between checkpoint with restart and replication

A. Checkpoint with restart for P processors executing communicating tasks

Consider a *single* processor with MTBF M. P processors thus have a MTBF of $M' = M/P$. Let X be the checkpoint interval. t_x and t_r are the time it takes to create a checkpoint, and recover from check point respectively. A job executes for T_s time if no failure occurs.

In [4] Daly constructed a detailed model of wall clock application execution time on a computer system that exhibits Poisson single component failures. In the model, execution time includes the time to perform checkpoints and the time to redo the work performed between the last checkpoint and a failure, i.e., rework time. By adapting the MTBF as a result of the use of P processors, we can modify Daly's model for long-running parallel applications, where the execution time (T) is:

$$T = \frac{M'}{P} e^{t_r/M'} (e^{(X+t_x)M'} - 1) \frac{T_s}{X} \text{ for } t_x \ll T_s$$

The speedup is thus defined as T_s/T. The optimal checkpoint interval X that maximize T_s/T can be approximated as follows,

$$X_{opt} = \begin{cases} \sqrt{2t_x M'} \left[1 + \frac{1}{3}(\frac{t_x}{2Mi'})^{\frac{1}{2}} + \frac{1}{9}(\frac{t_x}{2M'})\right] - t_x & t_x < 2M' \\ M' & t_x \geq 2M' \end{cases} \quad (1)$$

B. Replication for P processors executing communicating tasks

P processors are divided into $P/2$ logical processes. For each logical process, there are two replicas running in parallel. Consider the interval between the nth and $n+1$th failure point. The average length of the interval is

$t_x + t_r + \frac{2}{P\lambda}$. The average progress made in this interval by a single processor is $\frac{2}{P\lambda}$. Applying renewal theory, we have

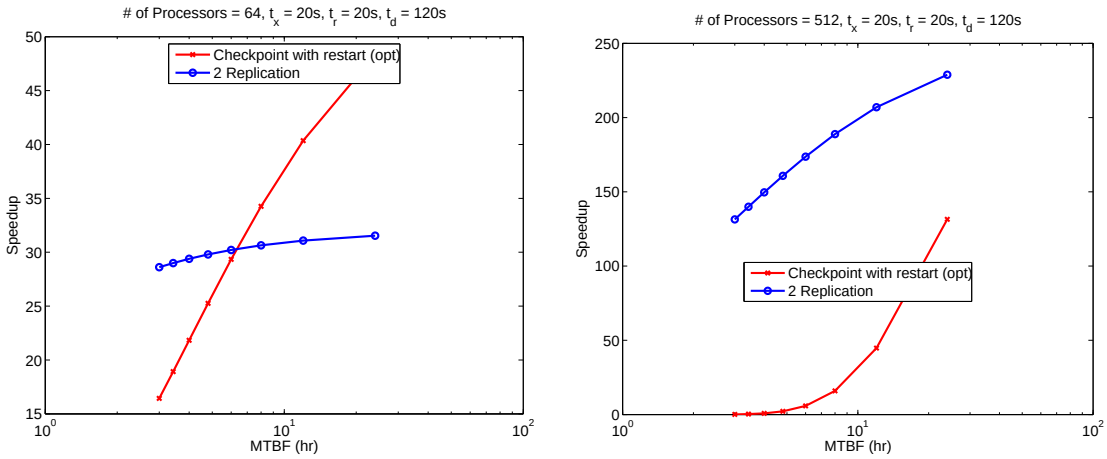
$$\overline{R} = \frac{\frac{P}{2}\frac{2}{P\lambda}}{t_x + t_r + \frac{2}{P\lambda}} = \frac{\frac{1}{\lambda}}{t_x + t_r + \frac{2}{P\lambda}}$$

Clearly, if $\frac{1}{\lambda} \gg t_x, t_r$, $\overline{R} \approx P/2$. Two assumptions are made in the above models. First, replicas do not fail at the same time. This is a reasonable assumption considering the MTBF of a single processor is high. If the replicas are chosen carefully such that highly correlated failure can be avoided (or in another work, independent failures can be assumed), the probability that replication fails is given by $(1 - exp(-\frac{t_x+t_r}{M})) \cdot (1 - exp(-\frac{t_r}{M}))$. For example, for $t_x = 300s$, $t_r = 600s$, $M = 1$ day, the probability is 7.17×10^{-5}. Secondly, we assume the normal process to wait till the failed process to recover from the failure. This assumption errors on the pessimistic side and simplifies the synchronization of the two replicas.

C. Numerical results

In this section, we compare numerical results under realistic failure behaviors. Eq. (1) is used to compute the optimal checkpoint interval under different settings. The speedup under checkpoint restart is then determined using the optimal checkpoint interval.

Fix P, varying MTBF: Here, $P = 64, 512$, $t_x = 300s$, $t_r = 600s$. MTBF varies from 1 day to 7 days (Figure 1).



(a) Number of processors = 64 (b) Number of processors = 512

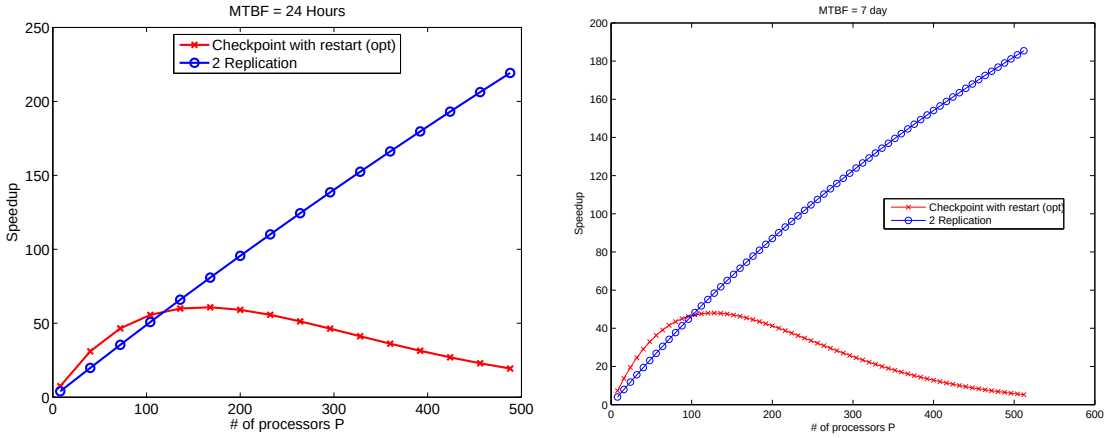
Fig. 1. Fix P, varying failure rate λ

Fix MTBF, varying number of processor P: Here, $t_x = 300s$, $t_r = 600s$, MTBF is set to be 1 day or 7 days. P varies from 8 to 512 (Figure 2). The basic observations from the numerical results are, which motivates the use replication in VolPeX, 1) little progress can be made by checkpoint with restart when the processor number grows large, 2) replication can effectively utilization the availability of abundant CPU resources.

IV. Research Tasks

We have identified the following research tasks in incorporating availability information in system design of VolPeX.

- More comprehensive trace data collections and models are needed for wide-area computing resource availability beyond host reachability. Currently, BOINC instruments BOINC clients to collect statistics such as ... This can be used to obtain large scale data and build better predication models.
- Performance of replication and checkpoint restart should be studied under more realistic settings. In particular, node heterogeneity is inherent among non-dedicated resources. Existing models typically assume homogeneous computing and bandwidth resources.
- Ultimately, we want the measurement and models can be used to guide better designs of VolPeX systems. They will be incorporated in the determination of system parameters, node selection and replica selection.



(a) MTBF = 1 day (b) MTBF = 7 day

Fig. 2. Fix MTBF, varying number of processor P

REFERENCES

[1] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. In *Proc. 2nd International Workshop on Peer-to-Peer Systems*, pages 256–267, 2003.

[2] J. Daly. A model for predicting the optimum checkpoint interval for restart dumps. *Lecture Notes in Computer Science*, pages 2660:3–12, 2003.

[3] J. Daly. A strategy for running large scale applications based on a model that optimizes the checkpoint interval for restart dumps. In *In Proceedings of the 26th International Conference on Software Engineering*, pages 70–74, 2004.

[4] J. Daly. A higher order estimate of the optimum checkpoint interval for restart dumps. In *Future Generation Computer Systems*, pages 22:303–312, 2006.

[5] A. Iosup, M. Jan, O. Sonmez, and D. Epema. On the dynamic resource availability in grids. In *Proc. of the 8th IEEE/ACM International Conference on Grid Computing (Grid 2007)*, pages 26–33, Austin, TX, USA, September 2007.

[6] D. Kondo, G. Fedak, F. Cappello, A. A. Chien, and H. Casanova. Characterizing resource availability in enterprise desktop grids. *Future Gener. Comput. Syst.*, 23(7):888–903, 2007.

[7] D. Kondo, M. Taufer, C. Brooks, H. Casanova, and A. Chien. Characterizing and evaluating desktop grids: An empirical study. 2004.

[8] J. W. Mickens and B. D. Noble. Exploiting availability prediction in distributed systems. In *Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design & Implementation*. USENIX Association, 2006.

[9] D. Nurmi, J. Brevik, and R. Wolski. Modeling machine availability in enterprise and wide-area distributed computing environments. In *Euro-Par*, 2005.

[10] K. Pattabiraman, C. Vick, and A. Wood. Modeling coordinated checkpointing for large-scale supercomputers. In *DSN '05: Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pages 812–821, Washington, DC, USA, 2005. IEEE Computer Society.

[11] B. Schroeder and G. A. Gibson. A large-scale study of failures in high-performance computing systems. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 249–258, Washington, DC, USA, 2006. IEEE Computer Society.

[12] M. Wu, X.-H. Sun, and H. Jin. Performance under failures of high-end computing. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, pages 1–11, New York, NY, USA, 2007. ACM.

[13] J. W. Young. A first order approximation to the optimum checkpoint interval. In *Communications of the ACM*, pages 17(9):530–531, 1974.